

Einführung in die Theoretische Informatik
Sommersemester 2020 – Übungsblatt 1

Ablauf der Tutorien

Die Tutoraufgaben werden *nicht* während den Tutorübungen gelöst. Bereiten Sie soweit wie möglich die Tutoraufgaben vor der Tutorübung vor und testen Sie ihre Kamera, Mikro und Möglichkeiten von **BigBlueButton** vor der ersten Tutorübung. Lesen Sie die Modalitäten auf der **Vorlesungswebsite** für mehr Informationen zur Tutorübung. Ihr*e Tutor*in wird Ihnen bekannt geben, wie Sie Fragen und Wünsche zur Tutorübung einbringen können.

AUFGABE 1.1. (*Wichtige Begriffe*)

Stufe A

Überprüfen Sie, dass Sie die Folgenden Begriffe korrekt definieren können.

- Alphabet Σ
- Wort w
- Länge eines Wortes
- leeres Wort
- Konkatenation von Wörtern
- Σ^+, Σ^*
- formale Sprache
- Operationen auf Sprachen (Def. 2.3)
- reflexive transitive Hülle
- Wortproblem
- DFA
- NFA
- akzeptierte Sprache
- Zustandsgraph
- Akzeptanzbedingung von DFAs/NFAs

AUFGABE 1.2. (*Automata Tutor*)

Stufe B

AutomataTutor ist ein online Übungstool, mit dem Studierende die Grundlagen der Automatentheorie erlernen können. Es bietet die Möglichkeit, viele der für die Vorlesung relevanten Probleme selbständig zu üben. In dieser Aufgabe geht es darum, dass Sie das Übungstool AutomataTutor kennenlernen. In den kommenden Wochen werden Sie über das Tool auch Hausaufgabenpunkte sammeln können.

- Rufen Sie die AutomataTutor Version der TUM auf. (<https://automata.model.in.tum.de/>)
- Loggen Sie sich mit Ihrer TUM-kennung (ab12cde) ein.
- Lösen Sie die Aufgaben unter ‘English to NFA’ und ‘English to DFA’, diese entsprechen der Aufgabe 1.6.

AUFGABE 1.3.

Stufe B

Sei $\Sigma = \{a, b\}$ und $W = \{aa, aaa, b\}$. Geben Sie, wenn möglich, jeweils mindestens drei Wörter an, die innerhalb bzw. außerhalb der folgenden Sprachen liegen.

- (a) $A = \{w \in W^* \mid |w| = 3\}$ (d) $D = \{w \in \Sigma^* \mid \exists u \in \Sigma^*. uw = w^2u\}$
(b) $B = \{w \in \Sigma^* \mid \exists u \in \Sigma^*. \exists v \in W. w = uv\}$ (e) $E = \{(ba^n b)^n \mid n \in \mathbb{N}_0\}$
(c) $C = \{w \in \Sigma^* \mid \exists n \in \mathbb{N}_0. |w|_a = n \cdot |w|_b\}$ (f) $F = \{w \mid w \in W^2 \wedge w \in W^3\}$

Hinweise:

- Mit $|w|_a$ bezeichnen wir die Anzahl der as in w .

AUFGABE 1.4.

Stufe B

Sei $\Sigma = \{a, b\}$, $A = \{\varepsilon, a, ab\}$ und $B = \{a, ba\}$. Wir betrachten nun verschiedene Verknüpfungen. Geben Sie für endliche Mengen alle Elemente und für unendliche drei Beispiele an:

- (a) AB (d) $A\emptyset$ (g) $\overline{B} := \Sigma^* \setminus B$
(b) A^2 (e) $A \times (\emptyset^*)$
(c) B^0 (f) $A\Delta B := (A \cup B) \setminus (A \cap B)$

AUFGABE 1.5. (*Endlich... Beweise!*)

Stufe D

Sei Σ ein Alphabet und $A, B, C \subseteq \Sigma^*$ beliebig. Entscheiden Sie, ob die folgenden Aussagen wahr oder falsch sind. Geben Sie für korrekte Aussagen einen Beweis an oder widerlegen Sie falsche mithilfe eines geeigneten Gegenbeispiels.

- (a) $A^* = A^+$ genau dann wenn (gdw.) $\varepsilon \in A$

- (b) $A(B \cap C) = AB \cap AC$
- (c) Falls $A \subseteq B$, dann $A^n \subseteq B^n$ für jedes $n \in \mathbb{N}_0$.
- (d) Unter der Annahme $A \neq \emptyset$ gilt: $A = AA$ gdw. $A = A^*$.

AUFGABE 1.6.

Stufe B

Geben Sie für jede der folgenden Sprachen über dem Alphabet $\Sigma = \{a, b, c\}$ einen DFA (graphisch) an, der genau die jeweilige Sprache erkennt. Um direkt Feedback zu Ihrer Lösung zu bekommen, bearbeiten Sie die Aufgaben in AutomataTutor.

- (a) Alle Wörter, die mit einem b beginnen.
- (b) Alle Wörter gerader Länge.
- (c) Alle Wörter ungerader Länge, die auf ein c enden.
- (d) Die Sprache, die nur das leere Wort enthält.
- (e) Alle Wörter, die **aab** enthalten.
- (f) Alle Wörter, die eine durch drei teilbare Anzahl von c enthalten.
- (g) $L = \{aabcaa, aacaa, baa\}$
- (h) Alle Wörter, deren 3-letzter Buchstabe ein a ist, z.B. **babb**.

Geben Sie nun für jede der folgenden Sprachen über dem Alphabet $\Sigma = \{a, b, c\}$ einen NFA an, der genau die jeweilige Sprache erkennt. Sie können diese Aufgaben ebenfalls in AutomataTutor bearbeiten.

- (i) Alle Wörter, die **aab** enthalten.
- (j) Alle Wörter, deren drittlezter Buchstabe ein a ist, z.B. **babb**. Der Automat sollte nicht mehr als 4 Zustände haben.

Beschreiben Sie dann in eigenen Worten, wie man im Allgemeinen einen Automaten konstruiert, der eine Sprache erkennt, die ...

- (a) die nur Worte enthält, die eine bestimmte Sequenz von Buchstaben enthalten
- (b) am Anfang/Ende jedes Wortes eine bestimmte Sequenz von Buchstaben fordert
- (c) nur Worte gerader/ungerader Länge enthält
- (d) von jedem Wort verlangt, eine bestimmte Anzahl an Buchstaben zu enthalten
- (e) deren Worte an einer fixierten Position einen bestimmten Buchstaben haben müssen.

AUFGABE 1.7.

Stufe B

Sei $\Sigma = \{a, b\}$ und $B_n := \{w \in \Sigma^* \mid \exists i : w_i = w_{i+n}\}$ die Sprache aller Wörter über Σ , in denen an irgendeiner Stelle der selbe Buchstabe im Abstand n vorkommt. Insbesondere ist B_0 die Menge aller nicht leeren Wörter, und B_1 die Menge aller Wörter in denen ein Buchstabe zweimal hintereinander vorkommt. *Versuchen Sie in beiden Aufgabenteilen NFAs und DFAs mit möglichst wenigen Zuständen anzugeben.*

- (a) Geben Sie jeweils einen NFA für B_0 , B_1 und B_2 an.
- (b) Geben Sie jeweils einen DFA für B_0 , B_1 und B_2 an.
- (c) Beschreiben Sie kurz, wie der DFA B_n und der NFA B_n für beliebige $n \in \mathbb{N}$ aussehen.
- (d) Beurteilen Sie die folgende Aussage: *Es gibt einen NFA für B_n mit $O(n)$ -vielen Zuständen, aber jeder DFA zu B_n hat mindestens $\Theta(2^n)$ -viele Zustände.*